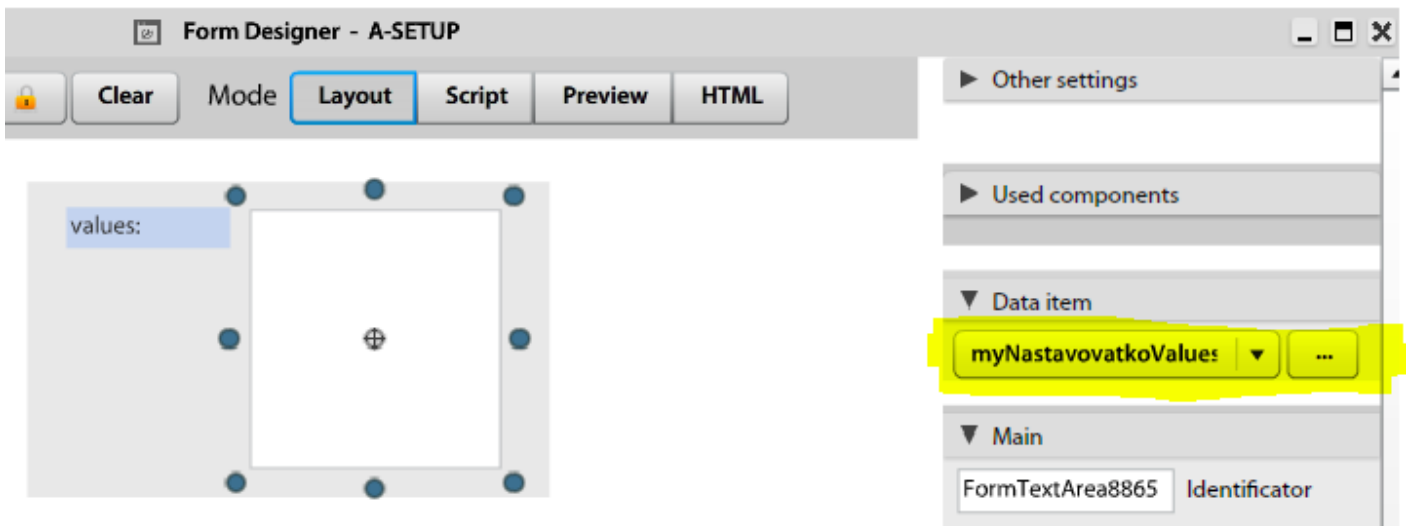


List Cross-form Values

1. Create Setup Custom Form

1a. Create your Setup form:

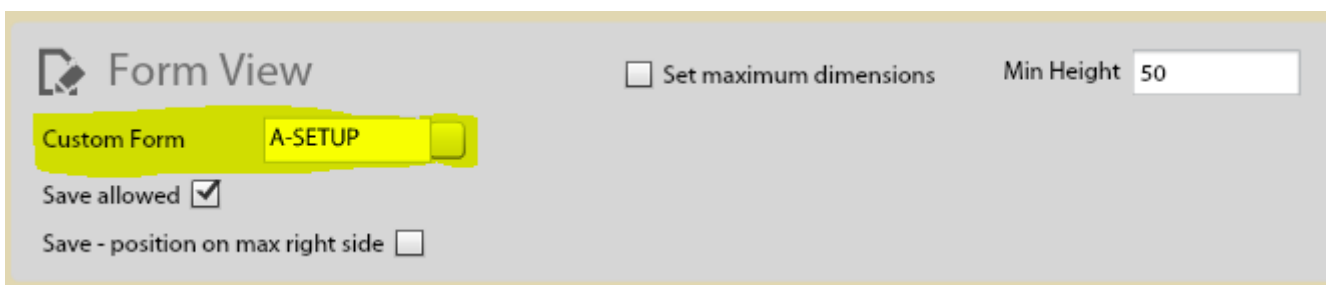


Note: use **TextArea** component as editor!

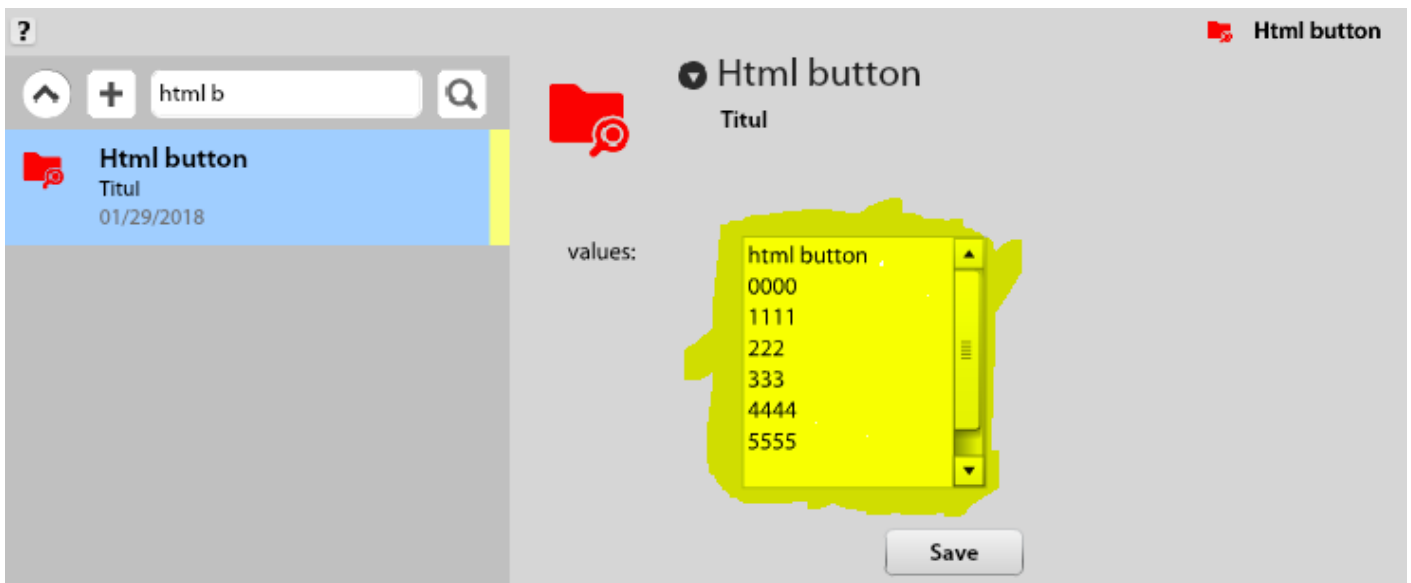
1b. Create and copy Data item to clipboard (i.e. myNastavovatkoValuesArea in this case)

2. Set Setup Custom Form

2a. Open ActivityPanel's Layout designer on your context, add **FormView** application and select our Setup form from *step 1*

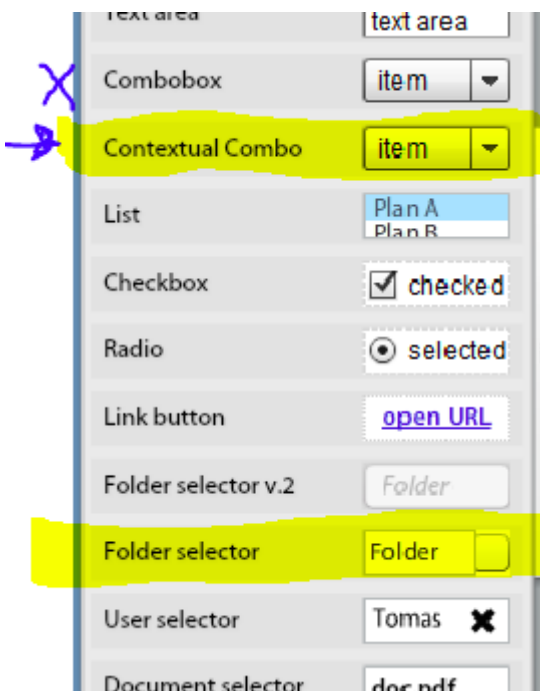


2b. Open your context and set some values (test case)



3. Create View Custom Form

WARNING: use **ContextualCombo** instead of basic Combobox! and Folder selector as Context selector

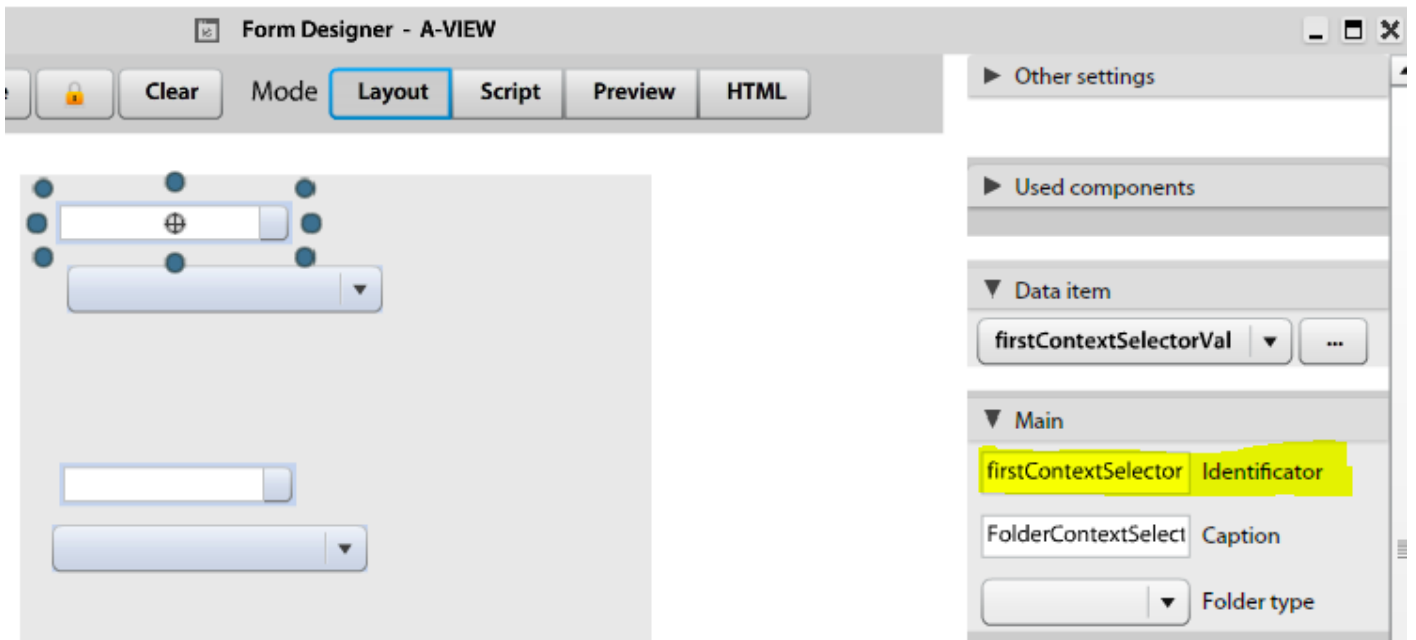


3a. Create your View form

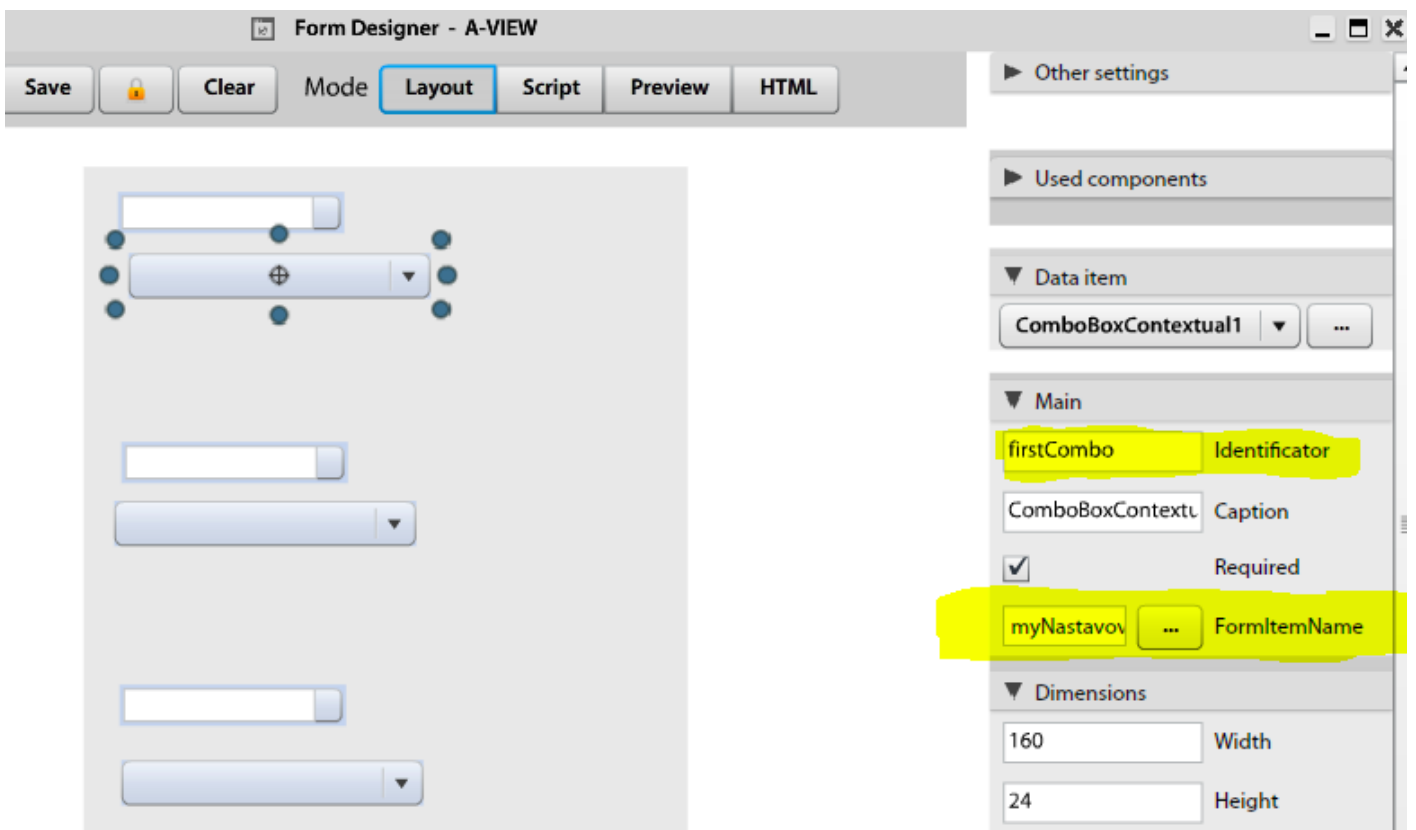
FYI: *ComboBoxContextual* get values from *context* (selected by Folder selector) by selected *FormItemName*.

Make pair Folder selector <-> ComboBoxContextual

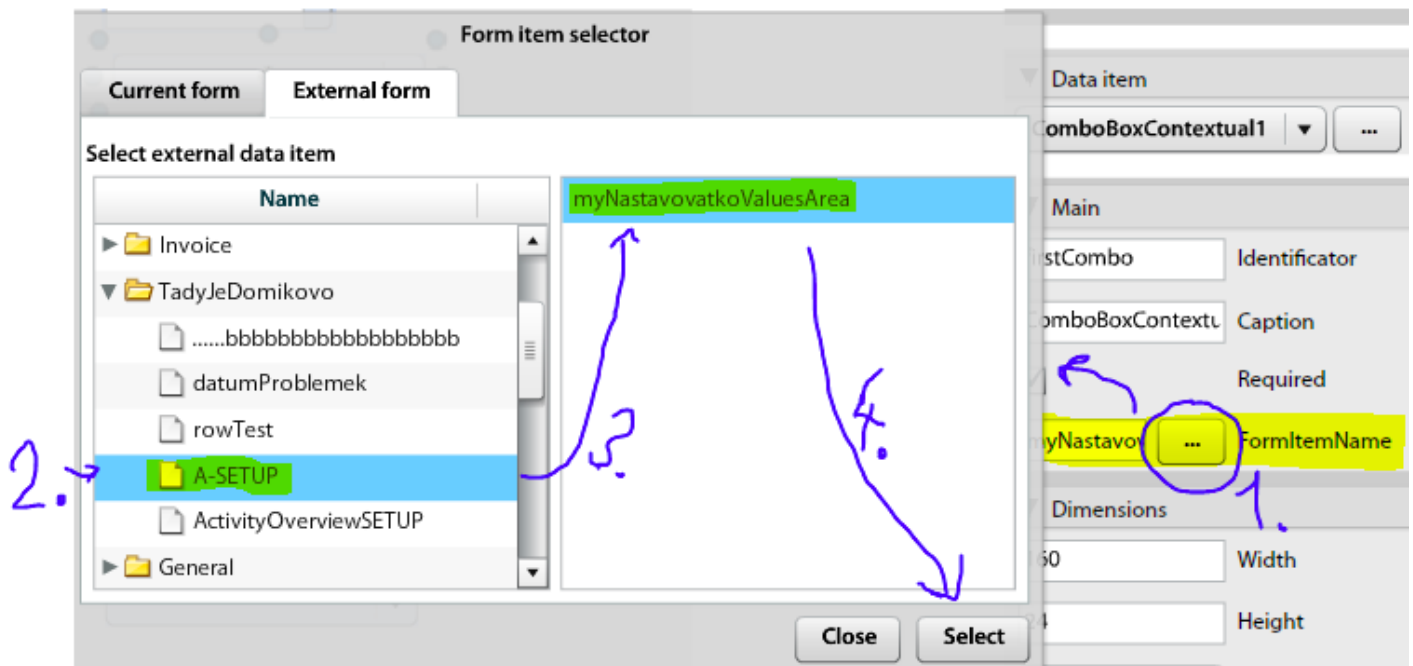
Folder selector setup: (**Identificator** required!)



ComboBoxContextual setup: (**Identificator** and **FormItemName** required!)

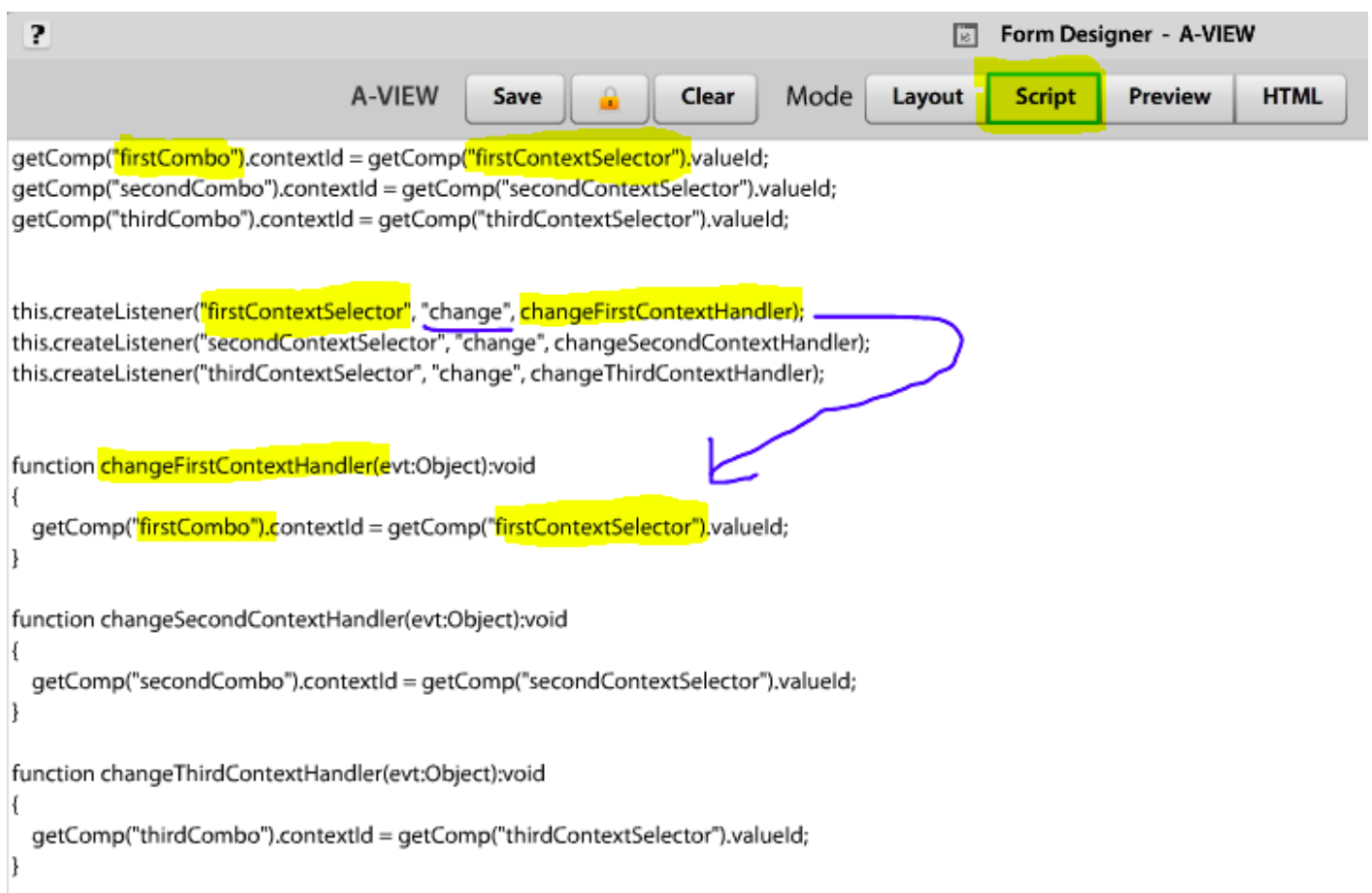


3b. Select **FormItemName** created in step 1b:



3c. Script setup (make combo-context pair)


Note: You can use one of **Ready-to-use Scripts** - check **attachments** (3context-3combo.txt in this case)



3d. Validate your script


4. Set View Custom Form

4a. Open ActivityPanel's Layout designer on your context, add **FormView** application and select our View form from *step 3*

 **Form View**

☐ Set maximum dimensions

Min Height

Custom Form **A-VIEW** 




Save allowed ☒


Save - position on max right side ☐


4b. Open your context:


(1) Select any context


(2) Select value from another form

  **Další tab** 

dalsi dialog 

ffff 

Html button 


html button 

html button


0000

1111

222

quality 

333

vy 

Done, **good job.**

- One context selector with one combobox

```
getComp("myCombo").contextId = getComp("contextSelector").valueId;

this.createListener("contextSelector", "change", changeContextHandler);

function changeContextHandler(evt: Object): void
{
    getComp("myCombo").contextId = getComp("contextSelector").valueId;
}
```

- One context selector with three comboboxes

```
getComp("firstCombo").contextId = getComp("contextSelector").valueId;
getComp("secondCombo").contextId = getComp("contextSelector").valueId;
getComp("thirdCombo").contextId = getComp("contextSelector").valueId;

this.createListener("contextSelector", "change", changeContextHandler);

function changeContextHandler(evt: Object): void
{
    var contextId: String = String(getComp("contextSelector").valueId);
    getComp("firstCombo").contextId = contextId;
    getComp("secondCombo").contextId = contextId;
    getComp("thirdCombo").contextId = contextId;
}
```

- Three context selectors with three comboboxes

```
getComp("firstCombo").contextId = getComp("firstContextSelector").valueId;
getComp("secondCombo").contextId = getComp("secondContextSelector").valueId;
getComp("thirdCombo").contextId = getComp("thirdContextSelector").valueId;

this.createListener("firstContextSelector", "change", changeFirstContextHandler);
this.createListener("secondContextSelector", "change", changeSecondContextHandler);
```

```
this.createListener("thirdContextSelector", "change", changeThirdContextHandler);
```

```
function changeFirstContextHandler( evt: Object): void
```

```
{
```

```
    getComp("firstCombo").contextId = getComp("firstContextSelector").valueId;
```

```
}
```

```
function changeSecondContextHandler( evt: Object): void
```

```
{
```

```
    getComp("secondCombo").contextId = getComp("secondContextSelector").valueId;
```

```
}
```

```
function changeThirdContextHandler( evt: Object): void
```

```
{
```

```
    getComp("thirdCombo").contextId = getComp("thirdContextSelector").valueId;
```

```
}
```

Revision #2

Created 15 March 2020 07:13:13 by Jan Safka

Updated 7 July 2021 12:50:35